CSC 261/461
Database Systems

Eustrat Zhupa

October 24, 2018

UNIVERSITY of ROCHESTER

XML

XML

- a tool for storing and transporting data.
- stands for eXtensible Markup Language
- a markup language much like HTML
- designed to store and transport data
- designed to be self-descriptive

XML

XML

- XML does not define any action or computation
- Consider this note to Tony from Jane, stored as XML:

```
<note>
  <to>Tony</to>
  <from>Jane</from>
  <heading>Reminder</heading>
  <body>Don't forget our meeting!</body>
</note>
```

- a piece of software may send, receive, store, or display it:

```
Note
To: Tony
From: Jane

Reminder
Don't forget our meeting!
```

XML

XML is Extensible

- applications will work as expected even if new data is added (or removed).
- Imagine an application designed to display the original version of `note.xml` (`<to> <from> <heading> <data>`).
- a newer version of `note.xml` adds `<date>` and `<hour>`, removing `<heading>`.
- older version of the application will still work:

```
<note>
  <date>2015-09-01</date>
  <hour>08:30</hour>
  <to>Tony</to>
  <from>Jane</from>
  <body>Don't forget meeeting!</body>
</note>
```

Old Version

```
Note
To: Tony
From: Jane
Head: (none)
Don't forget meeeting!
```

New Version

```
Note
To: Tony
From: Jane
Date: 2015-09-01 08:30
Don't forget meeting!
```

XML

XML Tree Structure

- XML documents are formed as element trees.
- XML tree starts at a `root` element and branches from the root to child elements.
- All elements can have sub elements (child elements):

```
<root>
  <child>
    <subchild>...</subchild>
  </child>
</root>
```

XML

XML Prolog

- The following line is called the XML prolog:

```
<?xml version="1.0" encoding="UTF-8"?>
```

- The XML prolog is optional. If it exists, it must come first in the document.
- XML documents can contain international characters: ő, å, ø.
- To avoid errors, you should specify the encoding used, or save your XML files as UTF-8.
- UTF-8 is the default character encoding for XML documents.

XML

Opening and Closing Tag

- All XML elements must have a closing tag
- it is illegal to omit the closing tag.

```
<p>This is a paragraph.</p>
```

- exception for empty elements.
- XML tags are *case sensitive*.
- Opening and closing tags must be written with the same case.
- all elements must be *properly* nested within each other.
  - the tag opened most recently is always the next tag to close

XML

XML

- XML elements can have attributes in name/value pairs just like in HTML.
- In XML, the attribute values must always be quoted:

```
<note date="12/11/2007">
  <to>Tony</to>
  <from>Jane</from>
</note>
```

XML

Special Chars

- Some characters have a special meaning.
- If you place a character like "<" inside an XML element, it will generate an error.

```
<message>salary < 1000</message>
```

- To avoid this error, replace the "<" character with an entity reference:

```
<message>salary &lt; 1000</message>
```

- There are 5 pre-defined entity references:

```
&lt; < less than
&gt; > greater than
&amp; &  ampersand
&apos; ' apostrophe
&quot; "  quotation mark
```

XML

XML Element

- An XML `element` is everything from the start tag to the element's end tag.

  ```
  <price>29.99</price>
  ```

- An element can contain:
  1. text
  2. attributes
  3. other elements
  4. or a mix of the above

XML

XML Naming Rules

- XML elements must follow these naming rules:
  - names are case-sensitive
  - names must start with a letter or underscore
  - names cannot start with the letters 'xml' (or XML, or Xml, etc)
  - names can contain letters, digits, hyphens, underscores, and periods
  - names cannot contain spaces
  - Any name can be used, no words are reserved (except `xml`).

XML

XML

```
<bookstore>
  <book category="children">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

XML

Extensibility

- XML elements are extensible
  - can be extended to carry more information.

```
<note>
  <to>Tony</to>
  <from>Jane</from>
  <body>Don't forget meeeting!</body>
</note>
```

```
MESSAGE
To: Tony
From: Jane
Don't forget meeeting!
```

```
<note>
  <date>2008-01-10</date>
  <to>Tony</to>
  <from>Jane</from>
  <heading>Reminder</heading>
  <body>Don't forget meeeting!</body>
</note>
```

Quotes

Attribute values *must* always be quoted. Either single or double quotes can be used.

- For a person's gender, the `<person>` element can be written like this:

  ```
  <person gender="female">
  ```

  or like this:

  ```
  <person gender='female'>
  ```

- If the attribute value contains double quotes you can use single quotes:

  ```
  <gangster name='George "Shotgun" Ziegler'>
  ```

  or you can use character entities:

  ```
  <gangster name="George &quot;Shotgun&quot; Ziegler">
  ```

Elements vs. Attributes

- Elements can be converted to attributes:

```
<person gender="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>

<person>
  <gender>female</gender>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

- Both examples provide the same information.

XML

Name Conflicts

- This XML carries HTML table information:

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

- This XML carries information about a table (a piece of furniture):

```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

- Both XMLs contain a `<table>` element, but with different content and meaning.
- A user/application will not know how to handle these differences.

Name Conflicts

- Name conflicts in XML can be avoided using a name prefix.

```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

XML

Well Formed XML Documents

- An XML document with correct syntax is called "Well Formed".
- The syntax rules were described before
  - XML documents must have a root element
  - XML elements must have a closing tag
  - XML tags are case sensitive
  - XML elements must be properly nested
  - XML attribute values must be quoted

XML

Valid XML Documents

- A "well formed" XML document is not the same as a "valid" XML document.
  - A "valid" XML document must be well formed.
  - In addition, it must conform to a document type definition.
- There are two different document type definitions:
  1. DTD - The original Document Type Definition
  2. XML Schema - An XML-based alternative to DTD
- A document type definition defines the rules and the legal elements and attributes for an XML document.

XML

DTD

- A DTD defines the structure of an XML document.
- It defines the structure with a list of legal elements:

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>

!DOCTYPE note defines that the root element of the document is 'note'
!ELEMENT note defines that 'note' element must contain the elements:
         "to, from, heading, body"
!ELEMENT to defines the 'to' element to be of type "#PCDATA"
!ELEMENT from defines the from element to be of type "#PCDATA"
!ELEMENT heading defines the heading element to be of type "#PCDATA"
!ELEMENT body defines the body element to be of type "#PCDATA"
#PCDATA means parse-able text data.
```